

Available online at www.sciencedirect.com

SCIENCE @ DIRECT®

Electronic Notes in Theoretical Computer Science

**Electronic Notes in
Theoretical Computer
Science**www.elsevier.com/locate/entcs

Characterizing polytime through higher type recursion

Karl-Heinz Niggl

Abstract

It is well-known that by a single use of higher type recursion on notation one can define proper Kalmar-elementary functions. This is due to a certain non-linear use of the ‘previous function’ in the step term of the recursion.

It is shown how to restrict recursion on notation in all finite types so as to characterise the polynomial time computable functions. The restrictions are obtained by enriching the type structure with ! s, and by adding linear concepts to the lambda calculus.

For this purpose, a simply typed term system RNA in the style of Gödel’s T is presented, where RNA indicates the essential restrictions to the formation of terms: Relevance, Necessitation, and Affinability. RNA-terms are built from variables and the ‘binary’ constants 0, s₀, s₁, p by restricted application accounting for R, by a simplification of Prawitz’s rule for introducing ‘!’ in his formulation of S4 accounting for N, by elimination of ‘!’, by restricted lambda abstraction accounting for A, and by rules for introducing case analysis and recursion on notation in all finite safe types. Safe types are those types which do not contain ‘!’. Similar to Gödel’s T, terms are interpreted over the set-theoretical function spaces.

Affinability formalises the underlying concept of linearity which is central to the whole system. It is designed such that it is decidable whether or not a given general term is in RNA.

It is shown that RNA is closed under reduction. Strong normalisation for RNA and uniqueness of normal forms then follow. In particular, every closed RNA term of ground type reduces to a numeral denoting the value of the term. Thus, RNA can be considered a simple functional programming language where computation is normalisation.

As for the expressive power of RNA, a kind of sharing concept implemented within RNA is employed to prove that every RNA program computing a number theoretical function can be evaluated in time polynomial in the lengths of the given inputs. That all polynomial time computable functions are definable in RNA simply holds by embedding the BC functions. (Joint work with S.Bellantoni and H.Schwichtenberg).
